T-ReX: a graph-based filament detection method

Tony Bonnaire, Nabila Aghanim, Aurélien Decelle, Marian Douspis

A&A 637, A18 (2020)
https://doi.org/10.1051/0004-6361/201936859

Astronomy & Astrophysics

T-ReX: a graph-based filament detection method

Tony Bonnaire[1,2], Nabila Aghanim[1], Aurélien Decelle[2], and Marian Douspis[1]

[1] Université Paris-Saclay, CNRS, Institut d'astrophysique spatiale, 91405 Orsay, France
e-mail: tony.bonnaire@ias.u-psud.fr
[2] Université Paris-Saclay, CNRS, Laboratoire de Recherche en Informatique, 91405 Orsay, France

Received 4 October 2019 / Accepted 17 February 2020

ABSTRACT

Numerical simulations and observations show that galaxies are not uniformly distributed in the universe but, rather, they are spread across a filamentary structure. In this large-scale pattern, highly dense regions are linked together by bridges and walls, all of them surrounded by vast, nearly-empty areas. While nodes of the network are widely studied in the literature, simulations indicate that half of the mass budget comes from a more diffuse part of the network, which is made up of filaments. In the context of recent and upcoming large galaxy surveys, it becomes essential that we identify and classify features of the Cosmic Web in an automatic way in order to study their physical properties and the impact of the cosmic environment on galaxies and their evolution. In this work, we propose a new approach for the automatic retrieval of the underlying filamentary structure from a 2D or 3D galaxy distribution using graph theory and the assumption that paths that link galaxies together with the minimum total length highlight the underlying distribution. To obtain a smoothed version of this topological prior, we embedded it in a Gaussian mixtures framework. In addition to a geometrical description of the pattern, a bootstrap-like estimate of these regularised minimum spanning trees allowed us to obtain a map characterising the frequency at which an area of the domain is crossed. Using the distribution of halos derived from numerical simulations, we show that the proposed method is able to recover the filamentary pattern in a 2D or 3D distribution of points with noise and outliers robustness with a few comprehensible parameters.

Key words. large-scale structure of Universe – methods: data analysis – methods: numerical – methods: statistical

1. Introduction

Large galaxy surveys like the Sloan Digital Sky Survey (SDSS, York et al. 2000) have confirmed the pattern drawn by matter at very large scales, which was initially addressed in analytical works and the first N-body simulations (e.g. Zel'dovich 1970; Doroshkevich & Shandarin 1978) and which has also been exhibited in early observations (see e.g. Joeveer et al. 1978; Einasto et al. 1980). In a pattern that is commonly referred to as the Cosmic Web (Bond et al. 1996), filaments act like cosmic highways, linking together large overdensities of matter and playing a key role in the dynamics of the universe. Since these early observations, the community has considerably enhanced the quality and the resolution of simulations with, for example, Millenium (Springel et al. 2005), Illustris (Vogelsberger et al. 2014), and Horizon-AGN (Dubois et al. 2014). These high-resolution simulations of dark matter (DM) evolution, which sometimes even include baryonic matter, have thus led to a more accurate spatial distribution of matter and allowed us to quantitatively characterise the different cosmic structures in terms of morphology, density, composition, etc. (see e.g. Colberg 2007; Aragon-Calvo et al. 2010a; Cautun et al. 2014; Gheller et al. 2016; Gheller & Vazza 2019). Revealing the faint filamentary pattern of the Cosmic Web in data often relies on the view of galaxies as tracers of the dark matter distribution and allows for the study of the influence of the cosmic environment on the formation and evolution of those tracers (e.g. Alpaslan et al. 2014a,b; Martinez et al. 2016; Kuutma et al. 2017; Malavasi et al. 2017, 2020; Laigle et al. 2018; Codis et al. 2018; Kraljic et al. 2020; Sarron et al. 2019). It usually involves either stacking or individual inspection of objects after their detection. The observation of the filamentary pattern is currently performed using different observables: X-ray emissions (see e.g. Dietrich et al. 2012; Eckert et al. 2015; Nicastro et al. 2018), weak lensing (e.g. Gouin et al. 2017; Epps & Hudson 2017), or through the Sunyaev-Zel'dovich effect (see e.g. Bonjean et al. 2018; Tanimura et al. 2019, 2020; de Graaf et al. 2019).

To perform such statistical and physical analyses, it is essential to detect the filamentary pattern in an automatic way and this task is even more challenging when dealing with real observations. Visual inspection makes it possible to easily identify the underlying structure, especially in mock datasets, whether we are dealing with the filament-like or clustered parts of the pattern. Over the years, the key question quickly has shifted to how we can automatically extract that which is visually observed. In 1985, Barrow et al. used, for the first time, a minimal spanning tree (MST; Boruvka 1926) approach in a cosmological context to exhibit the underlying filamentary pattern from a 2D or 3D galaxy distribution, arguing that the usual statistical procedures, such as the two-point correlation function, are not sensitive to this specific feature. Since then, several methods have been developed to analyse and describe this gigantic network and yet, filaments still have not been attributed with a unique, well-posed definition. In an intuitive way, filaments correspond to bridges of matter between two dense regions of the space. On the basis of this simple idea, many algorithms with their own mathematical definitions have emerged. With no aim of being exhaustive (see Libeskind et al. 2017, for a detailed review), we give hereafter a list of such methods for classifying cosmic web elements. Some are using the previously mentioned minimum spanning tree, an object coming from graph theory. The resulting tree highlights a preferable path minimising

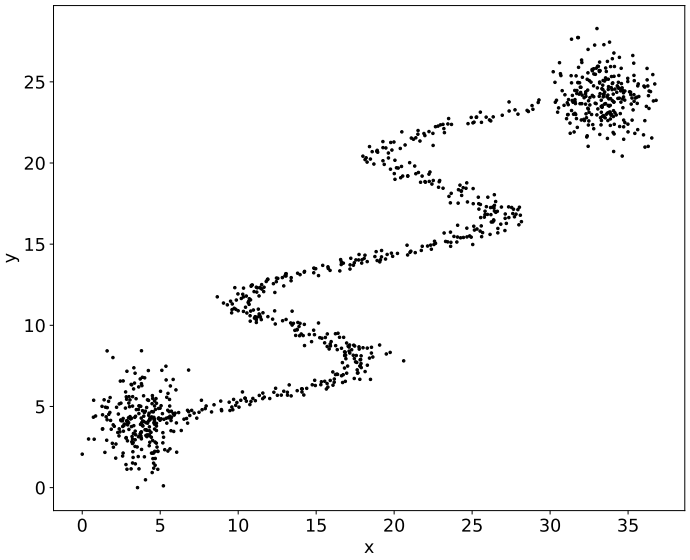

Fig. 1. Toy model used to illustrate steps of the algorithm corresponding to a rotated sinewave with Gaussian random noise linking two Gaussian clusters.

the total distance to link galaxies together (Barrow et al. 1985; Alpaslan et al. 2014a). After several processing stages of the graph proper to each method, filaments are extracted as branches of the tree. The study of the topological properties of the continuous density field through the Discrete Morse Theory led Aragón-Calvo et al. (2010b) and Sousbie (2011) to define filaments as the set of gradient lines linking maxima and saddle points. The seminal work of Aragon-Calvo et al. (2007) allowed Cautun et al. (2013) to build Nexus, an algorithm that performs a scale-space representation of the field in which filaments are defined locally through the relative strength between eigenvalues of the Hessian matrix of a smoothed continuous density obtained from the Delaunay Tessellation Field Estimator (Schaap & Weygaert 2000). Another class of methods is based on a statistical representation of stochastic point processes to model the geometry of the filamentary structure. In particular, Stoica et al. (2007) presented their modeling of filaments as connected and aligned cylinders through the marked point-processes theory. Genovese et al. (2014) and Chen et al. (2015) proposed that cosmic filaments be identified as ridges in the distribution of galaxies using an automatic algorithm moving iteratively a set of points along the projected gradient. Some indirect methods aim to first recover the initial density field and then make it evolve forward in time using the Lagrangian perturbation theory. Indeed, Kitaura (2013) and Jasche & Wandelt (2013), respectively, paved the way for Bos et al. (2014) and Leclercq et al. (2016) to develop such tools. We note that these methods are indirect reconstructions and are not specifically related to our issue of detecting cosmic web elements; although Leclercq et al. (2016) do use the inferred final density field in a game theory framework to classify structures in the reconstructed density field.

This wide variety of approaches, all aimed at identifying filaments in a spatial distribution of matter tracers, reveals how this problem can be hard to handle and also how great an importance it holds for observational cosmology. Also, some of the above methods are designed on simulations and using dark matter particles to detect those features but if we want algorithms to be able to handle real datasets, we need it to work specifically with galaxies (or halos in simulation) as inputs. With this in mind, we developed an algorithm using a set of 2D or 3D galaxy positions to build a smooth representation given by a graph structure and standing in the ridges of the distribution. The presented method does not rely on any density estimation but directly on the set of observed data points. It does not assume any shape for filaments but, rather, a global weak prior on Cosmic Web connectivity and can be easily extended to any topological prior as long as it is given by a graph structure. Furthermore, it can be used as a denoised representation of the Cosmic Web for other applications than filament detection.

In the first section, we present the datasets we use throughout this article to illustrate the steps and results of the proposed algorithm, called T-ReX (Tree-based ridge extractor). Section 3 provides the required mathematical formalism used to build the procedure. Section 4 develops the method step by step and illustrate the obtained results on a simple dataset, while Sect. 5 discuss the effect of each parameter on the resulting estimate of the underlying structure. Finally, Sect. 6 presents and discuss outputs obtained on cosmological datasets, then comparing it with other existing methods, namely Bisous, DisPerSE, and Nexus.

2. Data

In order to develop and test the main steps of the algorithm, we use a simple and non-cosmological dataset, hereafter called the toy dataset, shown in Fig. 1. It is constructed in a way so that it mimics a regularly curved structure, the filament, linking two clusters of points standing for overdense regions. The use of this toy dataset enables us to explore the impact of the parameters and test the reliability of the algorithm.

As a realistic cosmological dataset representing the Cosmic Web, we adopted the Illustris simulation outputs[1] (Vogelsberger et al. 2014). It is a set of large-scale hydrodynamical simulations with different resolutions in which an initial set of particles (dark matter or baryonic gas) distributed over a $75\,\mathrm{Mpc\,h^{-1}}$ box is evolved forward in time from high redshift to $z = 0$. From the resulting distribution at $z = 0$, halos of dark matter are identified using a Friend-of-Friend algorithm (FoF, More et al. 2011). To assess the application of the algorithm for cosmological cases and mimic its use for a galaxy survey, we consider structures inside halos, called subhalos, which have been identified with the Subfind algorithm (Springel et al. 2008) and provided by the Illustris package, as has already been done in other recent studies (Coutinho et al. 2016). For convenience, we sometimes refer to these subhalos as "galaxies". Figure 2 shows a thin $5\,\mathrm{Mpc\,h^{-1}}$ slice of dark matter distribution obtained from the Illustris-3 simulation in which subhalos have been extracted. We can see how these "galaxies" trace the underlying web drawn by the dark matter particles.

In the following, each time we use a dataset built from the Illustris simulation, it always concerns the box at redshift $z = 0$ and the Illustris-3 resolution obtained from 455^3 dark matter particles with a mass resolution of $4.0 \times 10^8\,M_\odot$. When needed, we will explicitly specify the settings with which the subset of particles is obtained. Namely, we will specify the type of particles we are showing (subhalos or DM particles), the cut in the spatial distribution (over x_e, y_e or z_e spatial axes), and the cut in total mass M over the considered particles in the spatial range.

Finally, to compare our results with other methods, we also apply T-ReX to FoF halos extracted from a $200\,\mathrm{Mpc\,h^{-1}}$ box of a Gadget-2 N-body simulation with 512^3 particles

[1] http://www.illustris-project.org/data/

(Springel et al. 2005). This particular simulation[2] is the one used in Libeskind et al. (2017), who proposed a unified comparison of the main existing procedures to classify elements of the cosmic web using either dark matter particles or dark matter halos as input.

3. General formalism

Relying on the simple and only assumption that observed points (i.e. galaxies) are tracing the underlying Cosmic Web, the main idea of T-ReX is to model the filamentary structure as the set of ridges (or principal curves) in the input point cloud. To extract these ridges, we use the minimum spanning tree and extend its previous application in cosmology (Barrow et al. 1985; Alpaslan et al. 2014b) by building a smooth version of it standing "in the middle" of the cloud. We note that this problem of finding curves passing through data points or detecting ridges in images is not proper to the cosmology field and has also been extensively studied in applied mathematics; it is currently of importance for medical applications, such as blood vessels segmentation (see e.g. Moccia et al. 2018, for a recent review) or dimensionality reduction (Qiu et al. 2017).

The basic idea behind this approach is that the true filamentary structure is a continuous manifold that can be described with a graph structure, while the observed galaxies represent a sparse and noisy sampling of that manifold. More precisely, in this paper is aimed at finding the best 1D representation of that manifold using a tree topology. This section introduces the required formalism to highlight how clustering methods as Gaussian Mixture Models (GMM), combined with graph theory, can be used to build such a representation starting from a general set of N datapoints $X = \{x_i\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^d$.

3.1. Elements from graph theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, with $\mathcal{V}$ as the collection of vertices, $\mathcal{E} = \{(i, j) \,|\, (i, j) \in \mathcal{V}^2\}$ as the set of edges linking nodes together, and $\{w_{ij}\}_{(i,j)\in\mathcal{V}^2}$ as the set of edge weights, such that $\forall (i, j) \in \mathcal{V}, w_{ij} \geq 0$. In our case, we consider $w_{ij} = \|x_i - x_j\|_2^2$. Let us also define d_i the degree of a node $i \in \mathcal{V}$ as the number of edges directly connected with it.

We call minimum spanning tree $\mathcal{M}$ the subgraph of $\mathcal{G}$ with $|\mathcal{V}| - 1$ edges that is reaching all nodes of $\mathcal{V}$ with the minimum total weight. By construction, $\mathcal{M}$ has no loops and is unique if there are not two edges with the same weight in $\mathcal{G}$, which, in our case, does not seem likely to happen since it would imply galaxies with the exact same distance between them. Still, it would only create very local modifications of the tree structure that would be erased by future operations. In a tree-like structure, we can define three exclusive typologies for a node i depending on its degree: extremity node ($d_i = 1$), junction node ($d_i = 2$), or bifurcation node ($d_i > 2$).

Graphs can be represented by some computable quantities encoding the full graph information. A first representation is given by the adjacency matrix of $\mathcal{G}$, noted $\mathbf{A}$, which is a symmetric $|\mathcal{V}| \times |\mathcal{V}|$ matrix encoding whether two vertices are linked or not. Elements A_{ij} of this matrix take values as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E}, \\ 0 & \text{if } (i, j) \notin \mathcal{E}. \end{cases} \quad (1)$$

[2] https://data.aip.de/projects/tracingthecosmicweb.html

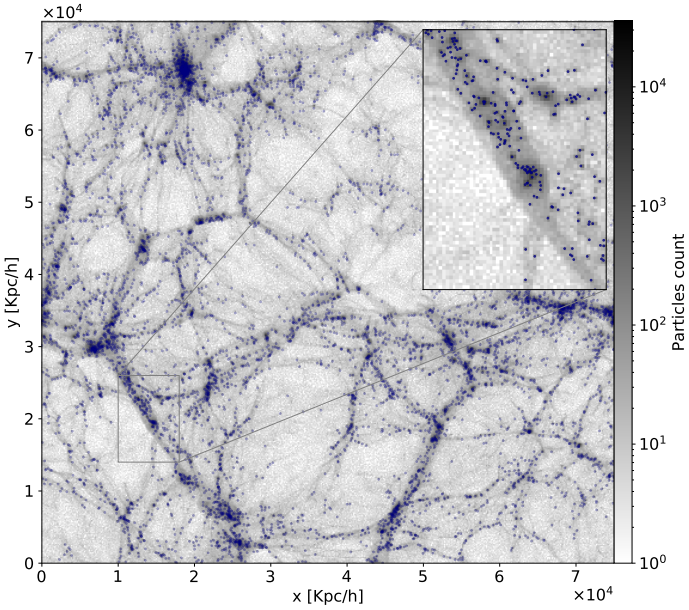

Fig. 2. Projected 2D slice ($z_e = [0; 5]\,\mathrm{Mpc\,h^{-1}}$) of dark matter particles distribution obtained from Illustris-3 simulation at a redshift $z = 0$ together with 2D projection of Subfind subhalos in the same region (blue dots).

This matrix encodes all the knowledge about the connectivity of vertices in the graph $\mathcal{G}$, and if we consider the matrix $\boldsymbol{W}$, such that $W_{ij} = w_{ij}A_{ij}$, we end up with a matrix describing the full graph.

Another useful representation of a graph is the Laplacian matrix from which spectral decomposition gives fundamental information about the graph structure (Lurie 1999). Let $\mathcal{G}$ be an undirected simple graph with an adjacency matrix $\boldsymbol{A}$ and $\boldsymbol{D}$ is a diagonal $|\mathcal{V}| \times |\mathcal{V}|$ matrix in which the element D_{ii} corresponds to the degree of the node i. Then the Laplacian matrix of $\mathcal{G}$ is the symmetric, positive semi-definite $|\mathcal{V}| \times |\mathcal{V}|$ matrix defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}. \quad (2)$$

As the MST reaches all data points, the resulting graph is not smooth and, therefore, it does not properly reveal the local geometry of the underlying distribution (see Fig. 3). In order to recover the shape of the distribution, we span the set of data points with a given number of centroids that will coarse grain the density distribution. This task is achieved by using Gaussian Mixture Models. The key idea of T-ReX, thus, is to achieve a smooth representation of the d-dimensional dataset standing in its ridges by computing a set of centroids with an enforced topology given by a graph structure.

3.2. Expectation-Maximization for Gaussian Mixture Models

Gaussian Mixture Models (GMM) are part of parametric mixture models that can be used to map a cloud of points to a density distribution by using a restricted number K of kernels to model the distribution. Starting with random parameters for Gaussian kernels, their positions and variances are adjusted iteratively to fit best the observed data. GMM are also extensively used in unsupervised clustering approaches where the aim is to partition the datapoints into K clusters by defining a probability that a given data point is part of the kth cluster. Using GMM, each cluster is represented by a Gaussian distribution and the clustering is reduced to an estimation problem of the Gaussian's parameters.

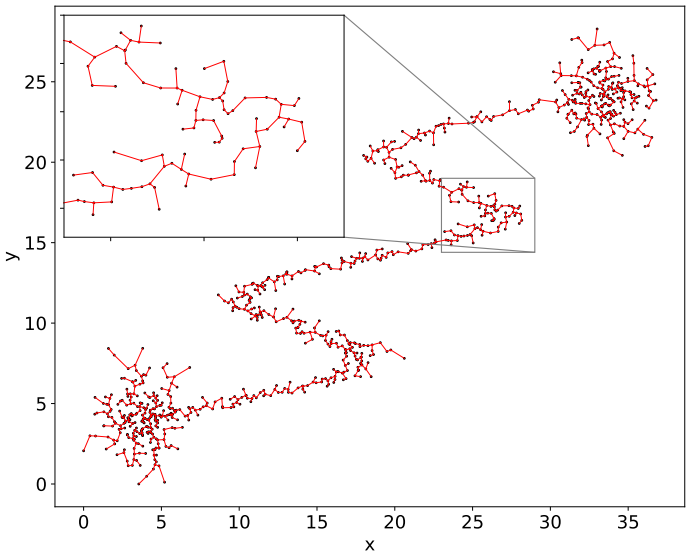

Fig. 3. Minimum spanning tree computed over data points of the toy dataset. Black dots are data points and straight red lines are edges of the tree.

Here we extend this second approach so that the clusters pave the observed set of datapoints in its ridges.

In practice, we define $K \leq N$ centroids $\{f_k\}_{k=1}^K$ with $f_k \in \mathbb{R}^d$ and assume that the dataset X is drawn in an independant and identically distributed way from an unknown density that we model as a weighted linear combination of K Gaussian clusters,

$$p(x\,|\,\theta) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x\,|\,f_k, \Sigma_k), \tag{3}$$

where $\theta = \{\pi_1, \ldots, \pi_K, f_1, \ldots, f_K, \Sigma_1, \ldots, \Sigma_K\}$ is the set of model parameters, π_k is the weight of the kth component, such that $\sum_{k=1}^K \pi_k = 1$, and $\mathcal{N}(x\,|\,f_k, \Sigma_k)$ is a multivariate normal distribution centered on f_k with covariance Σ_k.

This goal could also be achieved using the K-Means algorithm (Macqueen 1967) where we minimise the L_2 risk,

$$R[f] = \frac{1}{N} \sum_{i=1}^{N} \min_{k=1\ldots K} \|x_i - f_k\|_2^2. \tag{4}$$

This kind of similarity-based clustering of the data, however, generates a hard partition of the input domain, meaning that each point x_i can only be member of one group f_k and generally lacks of flexibility and robustness to noise and outliers. Mixture models can be used to face this difficulty by considering the conditional probability of a data point being part of a cluster given the assumed model.

From the assumption that the data are drawn from such a density, all we have to do is to estimate the values for θ fitting best the observed data. This is generally achieved by maximising the log-likelihood function,

$$\mathcal{L}(\theta; X) = \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_i\,|\,f_k, \Sigma_k)\Big), \tag{5}$$

from which, in this case, it is impossible to get an analytic solution when maximising with respect to θ.

To bypass this difficulty, we use an Expectation-Maximization (EM) approach (Dempster et al. 1977) by defining a set of latent variables, $Z = \{z_i\}_{i=1}^N$, encoding the partition of the dataset: $z_i \in [\![1, K]\!]$ denotes which of the K Gaussian components x_i belongs to. The completed log-likelihood is then

$$\mathcal{L}(\theta; X, Z) = \sum_{i=1}^{N} \log(\pi_{z_i} \, \mathcal{N}(x_i\,|\,f_{z_i}, \Sigma_{z_i})), \tag{6}$$

which can be maximised using EM approach.

As we introduced a new unknown quantity through Z, the central idea of the EM algorithm is to alternatively estimate Z by the expectation over $p(z\,|\,x)$ (E-step) and then update the parameters of the mixture θ by maximising the new likelihood on the basis of the current distribution for Z (M-step). This procedure provides an algorithm that locally maximises the true likelihood. Mathematically, the procedure can be understood more generally as follows; for any probability distribution over the latent variables, $q(Z)$, it reads,

$$\begin{aligned}\mathcal{L}(\theta; X) &= \sum_z q(z) \log\left(\frac{p(x, z\,|\,\theta)}{q(z)}\right) - \sum_z q(z) \log\left(\frac{p(z\,|\,x, \theta)}{q(z)}\right) \\ &= L(q, \theta) + D_{\mathrm{KL}}(q \,\|\, p(z\,|\,x, \theta)),\end{aligned} \tag{7}$$

where $D_{\mathrm{KL}}(q \,\|\, p) \geq 0$ is the Kullback-Leibler divergence (Kullback & Leibler 1951), implying that $L(q, \theta)$ is a lower bound for the log-likelihood.

The idea behind EM formalism is to maximise the lower bound $L(q, \theta)$ instead of the log-likelihood directly. The E-step consists of fixing θ and maximising $L(q, \theta)$ with respect to q. By noting that $\mathcal{L}(\theta; X)$ does not depend on q, we simply need the divergence to be cancelled out in order to maximise the lower bound and, thus,

$$q(z) = \operatorname*{argmax}_{q(z)} L(q, \theta) = p(z\,|\,x, \theta), \tag{8}$$

which can be computed using Bayes' theorem. In the M-step, considering we are performing the tth iteration, we fix $q(z) = p(z\,|\,x, \theta^{(t)})$ and update the optimal set of parameters, such that $\theta^{(t+1)} = \operatorname{argmax}_\theta L(q, \theta)$.

To summarise, EM is an iterative approach capable of identifying K clusters from the data itself with guaranteed convergence. In a first step (E), a probabilistic (soft) assignment of each data point to mixture components is computed and in a second one (M) an estimation of mixtures' parameters is performed given the distribution for the latent variables. The main advantage over the K-means method is that GMM allow a soft partitioning of the input dataset through this $q(z)$ distribution.

3.3. Regularised GMM for ridge extraction

So far, we have simply addressed the Gaussian mixture clustering with an Expectation-Maximization approach and gained access to K separated clusters, with their own means $\{f_k\}_{k=1}^K$ and covariances $\{\Sigma_k\}_{k=1}^K$ representing the data, but with no smoothness constraints or topology enforced. From the observation that the MST naturally traces ridges and the underlying connectivity of datapoints without any free parameters, we can enforce a tree topology to our centroids to obtain a representation that combines this idea of the MST and the local averaging naturally provided by GMM to impose smoothness. The question the full formalism tries to answer is what smooth minimal tree structure fits the set of observed data best. In general, if we want the centroids to have a given shape, we need to incorporate a prior distribution $p(\theta)$ within the previous equations. The presented framework is very close and inspired, in its form and

spirit, to manifold learning methods for dimensionality reduction (see e.g. Roweis & Saul 2000; Gorban & Zinovyev 2005) and, in particular, the principal curves (Hastie et al. 1989) field, which has already studied the application of mixture models to curve extraction from point distribution (Tibshirani 1992; Bishop & Svensén 1998).

With such a prior, we no longer aim to directly maximise the likelihood but the posterior $\log p(\theta \,|\, x) \propto \mathcal{L}(\theta; X) + \log p(\theta)$. In this context of maximum a posteriori estimation, previous equations and results from EM algorithm remains unchanged for the E-step, the maximization over q being independant on $p(\theta)$. In the case of the M-step, the update is computed so that that $\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, L(q, \theta) + \log p(\theta)$.

The log-prior can be considered as a regularization term on the log-likelihood and keeping in mind its role helps us choosing it correctly. In particular, we want to give centroids a smoothness constraint and to enforce a topology through a given graph structure $\mathcal{G}$. Hence, we use a Gaussian form for the prior with a variance ν^2 thus acting on the L_2 norm $\|\boldsymbol{F}\|^2_{\mathcal{G}}$ to constrain the smoothness of centroids directly on the graph domain, as is usually done in statistics (Smola et al. 2001) and which is inspired by previous studies on elastic topology regularization (Durbin & Willshaw 1987; Yuille 1990) and manifold learning (Gorban & Zinovyev 2005):

$$\begin{aligned}\log p(\theta) &= -\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K} b_{ij}\frac{\|f_i - f_j\|_2^2}{\nu^2} + \text{const.} \\ &= -\frac{1}{\nu^2}\mathrm{Tr}\{\boldsymbol{F}\boldsymbol{L}\boldsymbol{F}^T\} + \text{const.} \end{aligned} \tag{9}$$

where $\boldsymbol{F} \in \mathbb{R}^{d\times K}$ such that column k of $\boldsymbol{F}$ contains f_k and L is the Laplacian matrix as defined in Eq. (2).

In the context of this paper and its application, we simplify this formalism by considering equidistributed Gaussian mixtures ($\forall k \in [\![1, K]\!], \pi_k = 1/K$) with identical and isotropic covariances $\sigma^2\boldsymbol{I}_d$, where $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix. This reduces the problem with regard to the estimate of $\theta = \{f_k\}_{k=1}^K$ during the M-step. By noting $p_{ik} = p(z_i = k \,|\, x_i, \theta_k)$, the probability of a given data point x_i being well represented by the cluster k, we find

$$\begin{aligned}\theta^t = \underset{\theta}{\operatorname{argmax}} &-\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik}\frac{\|x_i - f_k\|_2^2}{\sigma^2} \\ &-\sum_{i=1}^{K}\sum_{j=1}^{K} b_{ij}\frac{\|f_i - f_j\|_2^2}{\nu^2}. \end{aligned} \tag{10}$$

The first term of this optimisation problem corresponds to a soft K-means clustering (Bezdek 1981) while the right-hand side is an elastic regularization term constraining the topology of centroids. Under the previous simplifications and in pursuit of a specific topology given by the minimum spanning tree, the presented formalism is equivalent to the work of Mao et al. (2015).

Again, to simplify the notation and to link the two variances σ^2 and ν^2, we can introduce the parameter $\lambda = \dfrac{\sigma^2}{\nu^2}$ as the relative strength of the two kernels. The final problem of the M-step can hence be written as

$$\begin{aligned}\theta^t = \underset{\theta}{\operatorname{argmin}} &\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik}\|x_i - f_k\|_2^2 \\ &+ \lambda\sum_{i=1}^{K}\sum_{j=1}^{K} b_{ij}\|f_i - f_j\|_2^2. \end{aligned} \tag{11}$$

The first term of this equation tries to minimise the error when datapoints are approximated by centroids while the second term acts like an elastic constraint on centroids when they are linked together in the considered graph. λ can be seen as a regularization parameter acting like a soft constraint on the total length of the graph and, thus, as a trade-off parameter between the data fidelity term and the penalty term constraining the smoothness and simplicity of the graph representation.

4. T-ReX: Tree-based Ridge eXtractor

Given a set of N observed data points $X = \{x_i\}_{i=1}^N$, each living in a d-dimensional euclidean space $\mathbb{R}^d$, the first step of T-ReX is to build a graph with a tree structure. This is achieved by computing the MST over X, resulting in a unique preferable path to link points together (see Fig. 3). This tree then goes through several processes to obtain a version that is robust to noise and outliers and to gain some smoothness properties.

4.1. Pruning of the tree

Considering that we obtained a graph with a tree structure, we adopt a simple denoising operation by cutting all the nodes standing in branches of the tree at a level l. In practice, branches are defined as the set of connected nodes linking an extremity node to a bifurcation node (defined in Sect. 3.1). Strictly speaking, we iteratively remove all nodes of degree one in the graph structure. By doing so, we remove the most spurious part of the structure corresponding to nodes that are more likely to be found in physically irrelevant regions for the underlying pattern (i.e. underdense regions). This approach is iterative, meaning that nodes which are initially bifurcations can become junctions or extremities (or even be removed if there are only branches with path length strictly lower than l connected to it). To give a representative image of this procedure, it acts like iterative peeling of an onion, attributing to each node a depth in terms of layers to peel before we reach it and starting from extremities (Hébert-Dufresne et al. 2016). This method is very close to the first step introduced by Barrow et al. (1985), where all branches with a path length inferior to l are removed (meaning that there are less than l nodes in the branch) except that our approach also cuts extremities of longer branches.

Previous MST methods usually perform, in addition to this pruning, a removal of all edges above a given physical length. In our case, this operation is not only aimed at avoiding the introduction of a new parameter that is not easy to tune, but it is also based on our argument that all connections, even "long" ones, can provide information about the underlying structure. Of course, as a result, if two unconnected parts of a network are given as an input to the presented method, they will end up connected.

Figure 4 shows the pruned MST obtained with a given cut-off level on the toy dataset. We can clearly observe that removing extremity nodes iteratively acts like a denoising operation, deleting small branches and irrelevant ones while preserving the core of the pattern. The choice of the pruning level is essential for a single realization of a tree, especially when dealing with noisy data. Section 5 analyses the impact of this parameter on the resulting tree.

4.2. The regularised minimum spanning tree

As discussed in Sect. 3, the MST does not exhibit a smooth behaviour. To enforce this constraint in our representation,

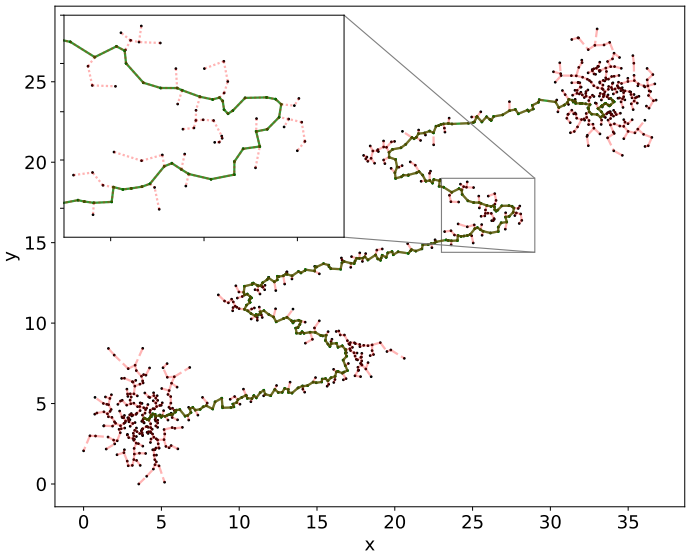

Fig. 4. Pruned version of the minimum spanning tree displayed on Fig. 3 at level $l = 28$. Black dots are data points, dashed shaded red lines are edges of the MST and green solid lines are the remaining edges after pruning.

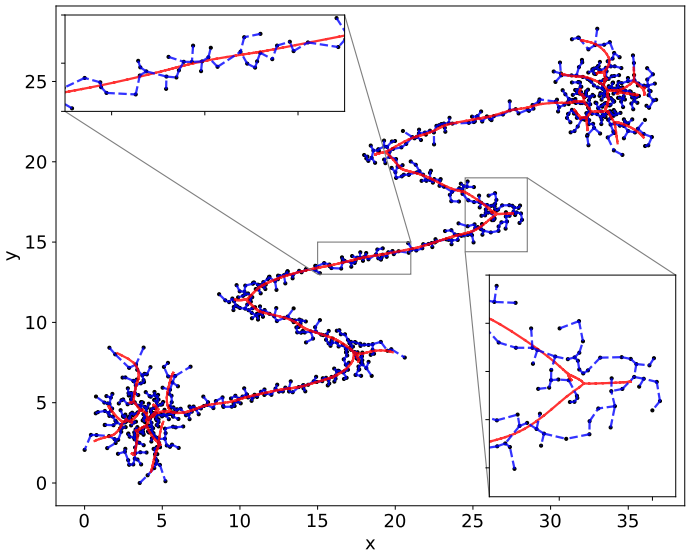

Fig. 5. Regularised minimum spanning tree computed over data points of the toy dataset. Black dots are data points, red solid lines are edges of the regularised tree and dashed blue line is the original MST. Result obtained from Algorithm 1 with $\lambda = 1$ and $\sigma^2 = 0.67$ (explained in Sect. 5).

we solve expectation-maximization Eqs. (8) and (11) following the work and notations of Mao et al. (2016) by applying Algorithm 1. It is worth noting that the inverse of $2\lambda\boldsymbol{L}+\boldsymbol{\Lambda}$ always exists since $\boldsymbol{L}$ is a positive semi-definite matrix and $\boldsymbol{\Lambda}$ is a positive diagonal matrix. The convergence is guaranteed by the EM approach and characterised by a slow displacement of the projected points $\|\boldsymbol{F}_t - \boldsymbol{F}_{t-1}\|_2^2 \leq \epsilon$ where t denotes the iteration index.

The computational complexity of Algorithm 1 can be divided into three components: (i) The computation of the MST over the centroids, (ii) The computation of the assignment matrix $\boldsymbol{P}$ to solve the E-step, and (iii) The matrix inversion to update centroids positions during the M-step. As already pointed out in Mao et al. (2015), the total complexity is $O(K^3 + DNK + K^2D)$.

Algorithm 1 Regularised minimum spanning tree

Input: Data: $\boldsymbol{X} \in \mathbb{R}^{d\times N}$, parameters: λ and σ
Output: $\boldsymbol{F} \in \mathbb{R}^{d\times K}$, the set of centroids and $\mathcal{B}$, the associated adjacency matrix

Initialise $\boldsymbol{F} = \boldsymbol{X}$ or with K-Means clustering

while convergence **do**
  Compute the minimum spanning tree $\mathcal{B}$ from $\boldsymbol{F}$
  Compute the Laplacian matrix $\boldsymbol{L}$ of $\mathcal{B}$ via Eq. (2)
  E-step:
    Compute the assignment matrix $\boldsymbol{P}$ where (i, k) entry is

$$p_{ik} = p(z_i = k \,|\, x_i, f_k) = \frac{\exp(-\frac{1}{2\sigma^2}\|x_i - f_k\|_2^2)}{\sum_{j=1}^{K}\exp(-\frac{1}{2\sigma^2}\|x_i - f_j\|_2^2)} \tag{12}$$

  M-step:
    Compute $\boldsymbol{\Lambda}$, a diagonal $K \times K$ matrix such that $\Lambda_{kk} = \sum_{i=1}^{N} p_{ik}$
    Solve Eq. (11) to update the position of centroids[3], $\boldsymbol{F} = \boldsymbol{X}\boldsymbol{P}(2\lambda\boldsymbol{L} + \boldsymbol{\Lambda})^{-1}$
end while

Figure 5 shows the difference between the MST directly built on data points and its regularised version obtained from Algorithm 1. The regularised minimum spanning tree (RMST) has smooth extensions (visible in the zooms of Fig. 5) while preserving the global shape of the tree-like structure. In the inflexion regions of the filament, we observe that the tree is a creating bifurcations. This is due to the chosen MST topology for the centroids. In this precise case, with a single filament, the best topology for centroids would be a straight line described by an adjacency matrix such that $A_{ij} = 2\delta_{i,i} - \delta_{i,j+1} - \delta_{i,j-1}$, where $\delta_{i,j}$ denotes the Kronecker delta function. It should be noted that such a topology could be handled by the formalism presented in Sect. 3.3.

4.3. The probability map

As previously mentioned in Sect. 3, a graph with a tree structure has no loops and, hence, it cannot represent holes but only, rather, connected components in the Cosmic Web topology. In addition to that, the MST highlights one particular path linking data points together but does not provide any idea of uncertainty or reliability of this latter. Both of these issues can be overcome by introducing a robust representation that takes into account the eventual variations in the input distribution. To do so, we build B different samples $\{X_b\}_{b=1}^{B}$ from the initial one X and compute the regularised MST for each of them in a similar fashion as in bootstrap approaches. The entire procedure is described by Algorithm 2.

From the B realisations of RMST, one can build a map $\boldsymbol{I}$ characterising the probability, in a frequentist meaning, of a

[3] In term of these matrices, optimization problem (11) can be written $\underset{\boldsymbol{F}}{\operatorname{argmin}}\, \mathrm{Tr}\{\boldsymbol{F}\boldsymbol{\Lambda}\boldsymbol{F}^T - 2\boldsymbol{X}\boldsymbol{P}\boldsymbol{F}^T + 2\lambda\boldsymbol{F}\boldsymbol{L}\boldsymbol{F}^T\}$.

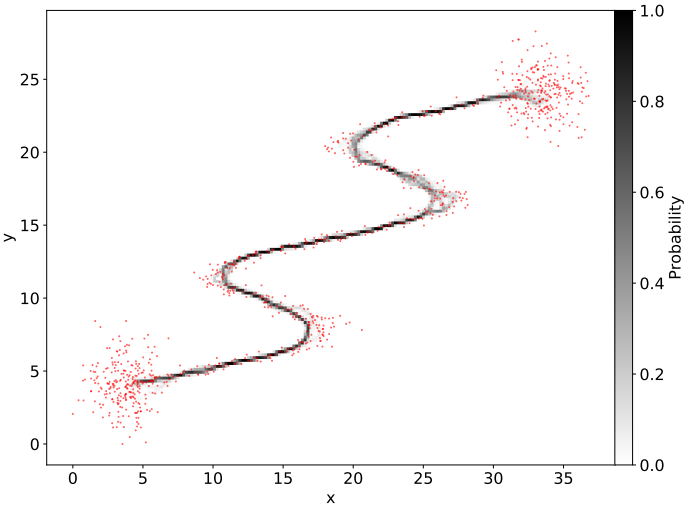

Fig. 6. Probability map obtained from Algorithm 2 and Eq. (13) with $B = 200$ and $N_B = 0.75N$. Red dots are input data points overplotted on the probability map.

position x to be crossed by a realization of a tree:

$$\boldsymbol{I}(x) = \frac{1}{B}\sum_{b=1}^{B} 1_{\boldsymbol{H}_b(x)=1}, \tag{13}$$

where 1_A is the indicator function and $\boldsymbol{H}_b$ is the binary histogram obtained from the projected points $\boldsymbol{F}_b$. The random nature of $\boldsymbol{I}$ thus comes from the uniformly at random resampling of X and not from Algorithm 1 that is a deterministic optimization step.

Algorithm 2 Bootstrap RMST

Input: Data $\boldsymbol{X}$, parameters $\lambda, \sigma, l, B, N_B$
Output: $\boldsymbol{S}$, the set of points describing the skeleton
Generate B bootstrap samples $\{\boldsymbol{X}^b\}_{b=1}^B$ of size N_B
for each $\boldsymbol{X}^b$ **do**
  Compute the MST $\mathcal{B}_b$ of $\boldsymbol{X}^b$
  Prune $\mathcal{B}_b$ at level l
  Keep the remaining vertices in $\mathcal{B}_b$, noted $\boldsymbol{Y}^b$
  Apply Algorithm 1 on $\boldsymbol{Y}^b$ with parameters λ and σ to obtain the regularised MST $\mathcal{B}_b^R$ and optimal $\boldsymbol{F}_b$
end for
$\boldsymbol{S} = \{\boldsymbol{F}_b\}_{b=1}^B$

Figure 6 shows a probability map obtained from the toy dataset in which the intensity of each pixel corresponds to the frequency that an edge of the MST crossed it. This way, we quantify the reliability of the various paths in the input domain. In practice, to build $\boldsymbol{I}(x)$, we use both the projected points $\boldsymbol{F}_b$ and the set of edges linking vertices encoded in $\mathcal{B}_b^R$ that contains information on the paths used and consequently should be taken into account in the final distribution. Edges are thus sampled and counted in the computation of $\boldsymbol{H}_b$ for Eq. (13). In what follows, we may refer to a quantity called the superlevel set of those maps defined as $\boldsymbol{\Gamma}_p(\boldsymbol{I}) = \{x \mid \boldsymbol{I}(x) \geq p\}$. Those sets are used to threshold the probability maps and keep only regions with a probability higher than p.

5. Choice of T-ReX parameters

In Table 1, we summarise the parameters of the algorithm together with their roles. We also give the baseline values further used in our study. As we are dealing with simulations of the Cosmic Web, we fix the cut-off level to a low value $l = 4$ and look for $B = 100$ regularised minimum spanning trees using uniformly at random 75% of the dataset for each sample. However, each of these parameters has a different and specific impact on the detection of the pattern that we discuss below.

5.1. Elastic constraint λ

As mentioned in Sect. 3.3, λ is a regularisation parameter acting like a trade-off between a set of centroids minimizing the data reconstruction error and the strength of the smooth tree topology we enforced. Hence, we understand that the larger λ, the more important the second part of Eq. (11), leading to a shorter and smoother tree, as seen on Fig. 7. λ can be seen as a soft-constraint on the total length of the tree, a high value leading to a tree representation that has short extensions and projected points are more uniformly distributed over the tree. Given the definition of λ in Sect. 3.3, it is also the ratio between both variances of Gaussian kernels we used, one for the data fitting term and the other for the prior on centroids to introduce the elastic regularization term. Choosing $\lambda = 1$ thus induces that the two kernels have the same variance. When dealing with outliers or highly noisy datasets, λ also helps increasing the robustness and maintains the tree structure in the desired regions without extending in noisy and underdense regions.

Mao et al. (2015) proposed to tune λ using the gap statistics, originally presented by Tibshirani et al. (2001) to choose the number of clusters in the K-means algorithm. This method requires several runs of the Algorithm 1 with a range of λ which can be very costly when dealing with large datasets. We hence choose to fix $\lambda = 1$ in our runs, leading to satisfactory results for a well chosen σ.

5.2. Spatial extension of Gaussian clusters σ^2

The parameter σ^2 corresponds to the variance of Gaussian clusters used to compute the assignment matrix $\boldsymbol{P}$ in Algorithm 1. It ensures the local smoothness of the graph by allowing a soft partitioning of the input data points into centroids. Thus, σ represents the spatial extension of each cluster and the higher it is, the more data points will be affiliated to a specific node of the resulting graph leading to a coarser representation. This trend is illustrated in Fig. 8, which shows several regularised MST obtained by fixing $\lambda = 1$ and varying σ. As σ increases, centroids tend to be aligned and they describe a coarser shape of the underlying structure, biasing the estimate. Intuitively, σ should represent the thickness of a typical filament so that centroids are fitting the distribution well.

To automatically tune this parameter from the data, we follow the recommendation of Chen et al. (2015), who investigated the choice of such a parameter in the SCMS algorithm. We thus chose σ using a modified version of the Silverman's rule (Silverman 1986):

$$\sigma_\mathrm{s} = A_0\Big(N(d+2)\Big)^{\frac{-1}{d+4}}\sigma_\mathrm{min}, \tag{14}$$

where A_0 is a constant, N is the number of data points, d is the dimension of the data and σ_min is the minimum standard deviation over all directions. Taking $A_0 = 1$ leads to the Silverman's rule and is the optimal estimate for an underlying Gaussian distribution. As argued by Chen et al. (2015), when the data are not Gaussian anymore, A_0 should be optimised as a free parameter. In our experiments, when the parameter is not explicitly defined,

Table 1. Parameters implied in the procedure and baseline values used in the presented results.

Parameter	Role	Used values
λ	Elastic constraint on centroids	1
σ^2	Spatial extension of Gaussian kernels	Eq. (14), $A_0 = 0.1$
l	The cut-off level to prune MST	4
B	Number of bootstrap samples	100
N_B	Size of bootstrap samples	$0.75\,N$

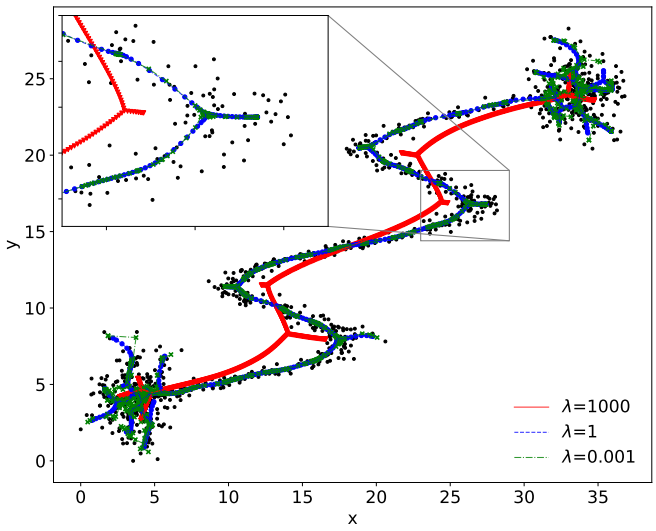

Fig. 7. Effect of the λ parameter on the regularised MST by fixing $\sigma = 1$. Black dots are data points while red, blue and green lines are RMST with respectively $\lambda = \{1000, 1, 0.001\}$. Projected points are also represented, respectively by triangles, dots and crosses. We note that curves for $\lambda = 1$ and $\lambda = 0.001$ are almost superimposed.

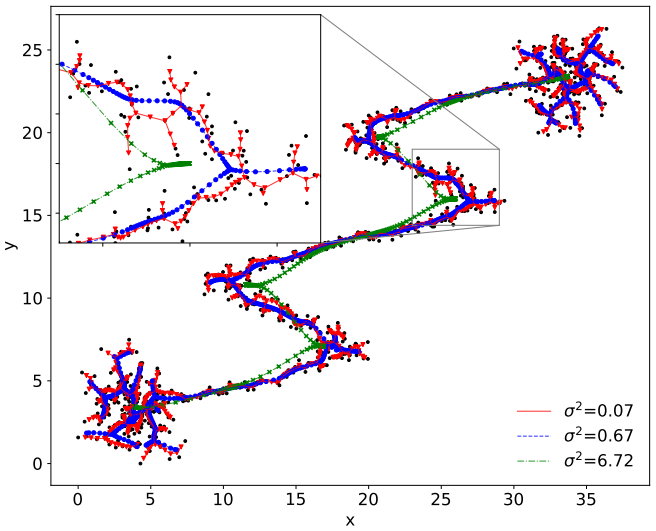

Fig. 8. Effect of the σ parameter on the regularised MST by fixing $\lambda = 1$. Black dots are data points while red, blue and green lines are RMST with respectively $\sigma^2 = \{\frac{\sigma_s^2}{10}, \sigma_s^2, 10\sigma_s^2\}$ (see Eq. (14)). Corresponding projected points are also represented, respectively, by triangles, dots and crosses.

we adopt the baseline value of Table 1, namely $A_0 = 0.1$, a rather low value so that the estimated trees keep some small scales variations. When A_0 increases, the smoothing scale also increases and a coarser filamentary pattern is described.

Although we considered a fixed isotropic and identical covariance matrix for all clusters, it is noteworthy that the formalism initially presented in Sect. 3.3 is more general. We could consider a specific covariance for each cluster, initialise it with the rule of Eq. (14) and adapt it automatically from the data. EM computation can indeed auto-adjust this estimate at each iteration by considering $\theta = \{f_1, \ldots, f_k, \Sigma_1, \ldots, \Sigma_k\}$ and then maximising the lower bound of the log-likelihood not only over f_k but also with respect to Σ_k in the M-step. This solution has, however, an additional computational cost and can lead each Gaussian cluster to be housed in a specific data point when K is close to N. It did not sufficiently improved the results in our cosmological application to consider it but could be included in future works. The current choice, hence, restricts the range of scales that can be described by the Gaussian clusters, implying that broad structures in which the extension is way above σ will not collapse into a single ridge passing in the middle of the structure in the resulting graph.

5.3. Pruning level *l*

As explained in Sect. 5.3, the pruning acts as a denoising operation but it also helps reducing the number of kernels to span the point cloud. A high cut-off level removes a large number of nodes at the extremity of all branches revealing only the core of the tree structure while a lower value allows branches to have long extensions reaching even nodes in empty regions. The choice of l can hence lead to different tree representations of a noisy dataset.

To choose the value of l, we rely on the work of Hébert-Dufresne et al. (2016) who introduced the onion decomposition for graphs. The idea is to attribute to each node a layer in terms of depth in the network allowing to define a center and a periphery. The left panel of Fig. 9 shows the onion spectrum of the noisy toy dataset and illustrates the points of the noisy dataset colored by their layers. The power-law decay in the first part of the spectrum can be interpreted as the removal of all short branches (in number of nodes). A constant level in the onion spectrum means that we are iteratively removing the same amount of nodes in the network at each iteration and thus that the tree structure is "stable" in terms of number of branches. Using as cut-off level the beginning value of the last constant level in the onion spectrum ($l = 39$ on left of Fig. 9) would lead to keeping only the core of the tree structure with a single branch (the longest one in the initial tree). However, this is a very conservative solution and doing so in real datasets would lead to miss some end parts of filaments or peripheral structures.

A threshold l that is too low can bring out spurious detections of the underlying pattern for a realization of a tree. However, this effect should be mitigated by:(i) the λ parameter which also helps reducing the length of branches in noise and outliers,

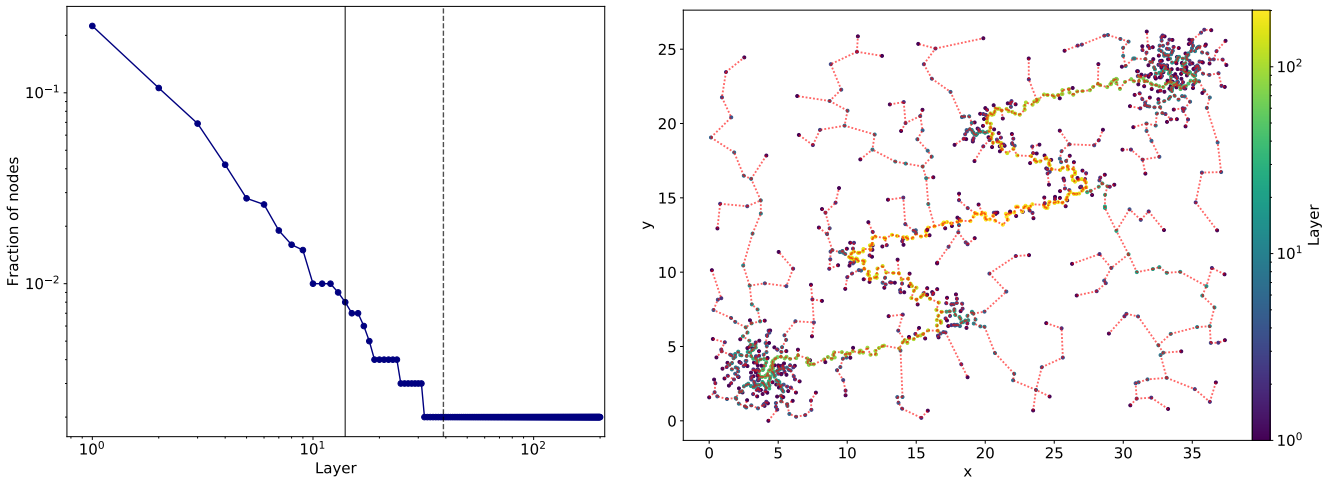

Fig. 9. *Left*: onion spectrum of the tree structure. Vertical lines correspond to $l = 14$ (solid) and $l = 39$ (dashed) discussed in Sect. 5.3. *Right*: layer value of each datapoint. Red dashed line is the MST and dots are data points from a noisy version of the toy dataset (obtained by adding 25% uniform noise in the bounding box).

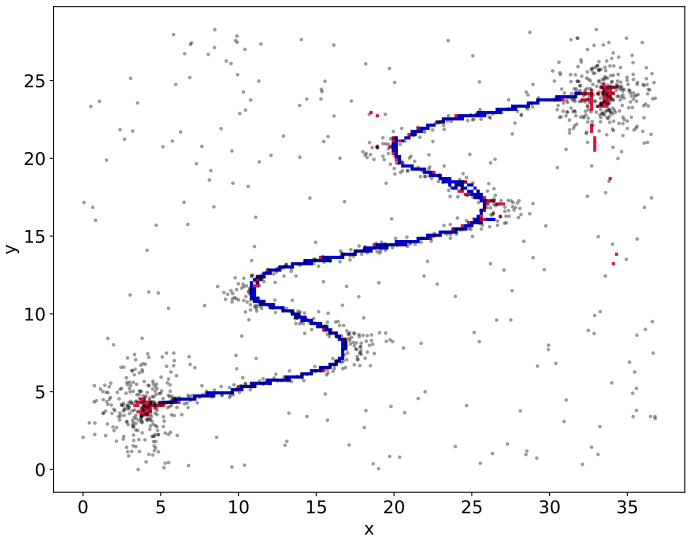

Fig. 10. Superlevel sets $\mathbf{\Gamma}_{0.25}(\boldsymbol{I})$ for two pruning levels on the noisy toy dataset: $l = 14$, a too low cut-off and $l = 39$, an adequate value. Blue pixels are regions where both sets are overlapping while red ones show regions highlighted by the $l = 14$ version but not by the $l = 39$ one and are mostly found in the background noise of the pattern.

as discussed in Sect. 5.1 and (ii) the bootstrap step where those detections will have a low occurrence as illustrated in the superlevel sets of Fig. 10. For this reason, in what follows, we consider a rather low value for the pruning parameter, namely $l = 4$. In the case of simulated cosmological datasets, this parameter only helps to remove data points that are located in empty or low dense regions since, for a well chosen value of σ, Algorithm 1 is robust to noise encountered around the ridges.

5.4. Number and size of the bootstrap samples

Both the number and size of the replicated samples, B and N_B respectively, are related to the probability map. Above a minimum value, the parameter B has almost no effect on the estimate for a fixed N_B. The main idea to explain this phenomenon is that, for a fixed size N_B, there is only a limited number of different possible paths with high probability. Even though a higher value for B can highlight some new paths, they will have a very low occurrence.

N_B affects the map in a more important way. A low size value induces more possible paths to cross and thus more variability in the resulting map while a size close to the initial one N ($0.90N$ for instance) allows for only local modifications of the highlighted paths; hence, it is more conservative. Choosing a low value can thus lead to more spurious path detection.

6. Results: application to cosmological datasets

In this section, we apply T-ReX with the baseline parameters of Table 1 on the 2D and 3D cosmological datasets described in Sect. 2. The slice of the 2D subhalo distribution corresponds to the data points in Fig. 2 which represent a projected slice of $5\,\mathrm{Mpc\,h^{-1}}$ depth. The 3D distribution of halos is built from a $200^3\,\mathrm{Mpc\,h^{-1}}$ Gadget-2 simulation used in Libeskind et al. (2017).

6.1. Filamentary structure in a 2D subhalo distribution

Figure 11 shows two realisations of a regularised minimum spanning tree (see Algorithm 1) with over 75% of the data points picked randomly and uniformly. Firstly, it is interesting to see that each RMST is standing in regions that would naturally be called ridges or filaments in the distribution of galaxies, that is, elongated structures connecting high density regions together. Secondly, we can see that according to the distribution of picked data points, different paths are taken for the core of the tree structure. The complementarity of the two realisations is highlighted in the zoomed region. Since the tree topology cannot include loops, the effect of disconnection is observable in this particular region where the solid blue realisation is not fully connecting the network. We note that such an effect is intensified by the pruning operation. Other realisations might not exhibit the disconnection in the same region as seen in the case of the dashed red line of Fig. 11. This highlights the necessity and the interest of stacking several RMST to obtain a full characterisation of the cosmic network.

Figure 12 shows a probability map obtained from 100 realizations of RMST. We can see that the highly probable part of the map is fitting what one would expect for the underlying

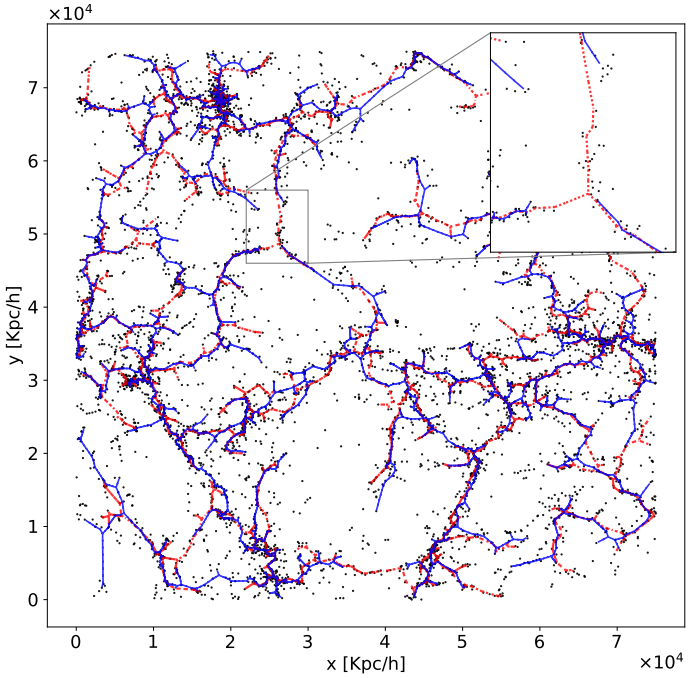

Fig. 11. Two realisations (solid blue line and dotted red line) of RMST (Algorithm 1) with 75% of the dataset picked randomly and uniformly with the parameters of Table 1. Black dots are subhalos from the Illustris-3 simulation.

distribution while the overlap of the superlevel set $\mathbf{\Gamma}_{0.25}(\boldsymbol{I})$ with the DM distribution allows us to see that high probability paths (above 0.25 in this case) are tracing the most prominent part of the network. It is worth noting that the agreement is particularly interesting given that the input of the algorithm are subhalos and not DM particles. The zoomed-in region emphasises that small scales are also recovered where high probability paths follow the ridge in the DM distribution.

6.1.1. Comparison with DisPerSE skeletons

DisPerSE (Sousbie 2011) is a publicly available[4] and widely used algorithm capable of detecting filaments and walls in a density field tracer, such as galaxy distribution. From this discrete set of particles, a continuous density field is estimated using the Delaunay tessellation field estimation. Based on the discrete Morse theory (Forman 1998), DisPerSE first aims at identifying singularities (or critical points) in the field defined as positions where the gradient cancels and then uses the local morphology to classify those points in maxima, minima, and saddles using eigenvalues of the Hessian matrix. DisPerSE finally identifies filaments using the connectivity of critical points following the gradient lines in the density field. Persistent homology (Edelsbrunner et al. 2002) is then used to remove insignificant parts of the pattern.

Figure 13 shows both T-ReX and DisPerSE results obtained on the Illustris slice with several probability thresholds for the former and several persistence levels for the latter. At fixed density smoothing (here 1), the DisPerSE skeletons show the best overlap with our un-tresholded probability map for a persistence $\sigma_p = 0$, where the two methods agree for most of the filamentary structure. The boundary effects observed in the DisPerSE

[4] http://www2.iap.fr/users/sousbie/web/html/indexd41d.html

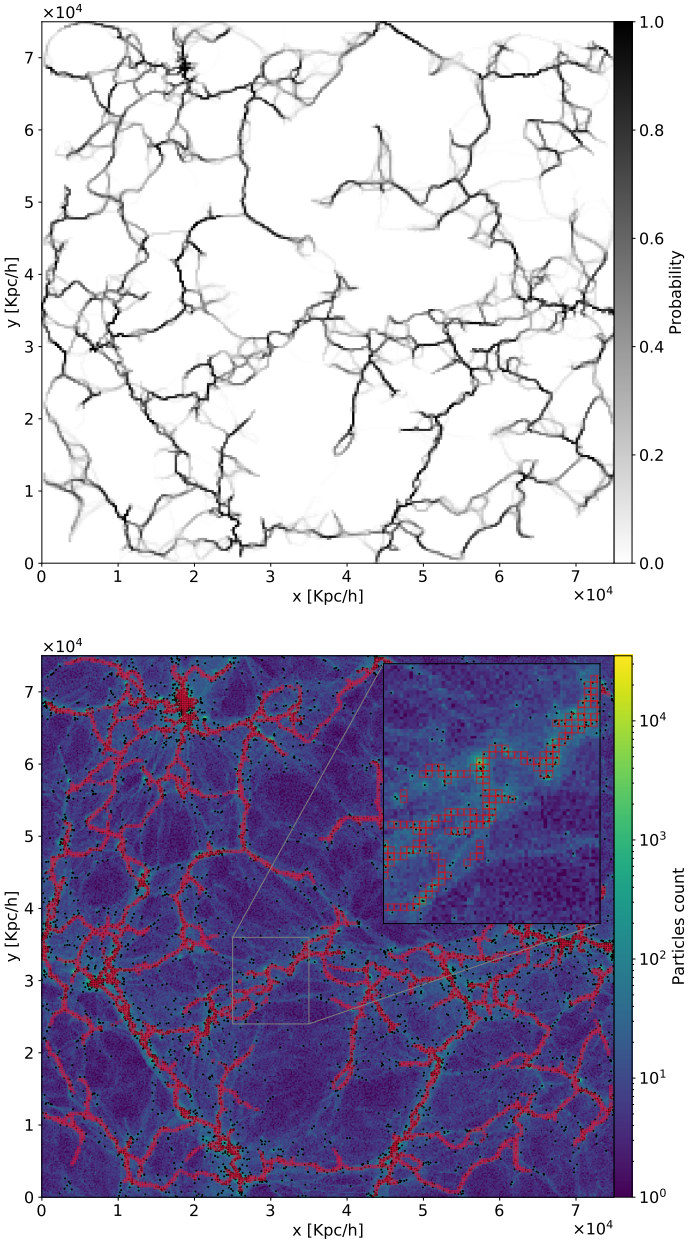

Fig. 12. *Top*: probability map $\boldsymbol{I}$ obtained from subhalos displayed in Fig. 2 with parameters described in Table 1. The resolution of the probability map is 250 Kpc h^{-1}. *Bottom*: superlevel set $\mathbf{\Gamma}_{0.25}(\boldsymbol{I})$ (red squares) overplotted on the DM distribution together with subhalos (black dots).

skeleton at low σ_p disappear with increasing persistence. The good agreement between high probability paths provided by T-ReX and the DisPerSE skeleton remains with increasing persistence levels and probability thresholds as shown by the overlap of DisPerSE and T-ReX skeletons (right column of Fig. 13). It should be emphasised that there is no direct transposition of the persistence threshold in DisPerSE into the probability threshold in T-ReX. The present choice of threshold parameters is hence arbitrary and only serves illustration purposes.

Although the two algorithms have very different definitions for what they both call filamentary pattern, it is reassuring to see that they are recovering similar structures. However, it is not surprising to observe some disagreement on specific filaments (see orange shaded regions in Fig. 13). Since the pattern identified by T-ReX is obtained by minimising a global criterion, some

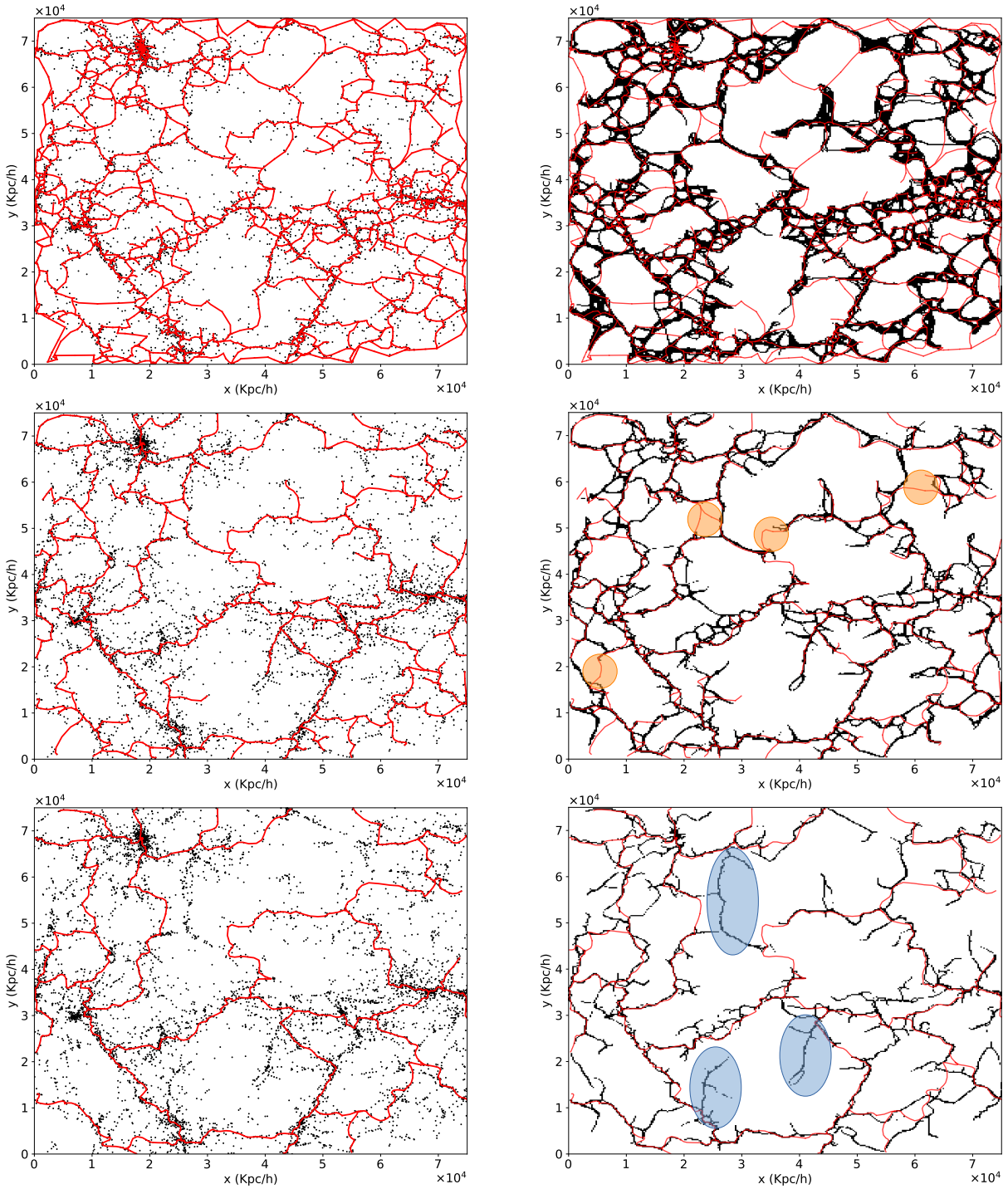

Fig. 13. *Left column*: subhalos (black dots) and DisPerSE skeletons (red lines) with several significance levels (*from top to bottom*: $\sigma_p = 0, 2, 5$). *Right column*: superimposition of some thresholded probability maps obtained by T-ReX and DisPerSE skeletons (red lines) with several significance levels (*from top to bottom*: $\sigma_p = 0$ and $\mathbf{\Gamma}_{0.0}(\boldsymbol{I})$, $\sigma_p = 2$ and $\mathbf{\Gamma}_{0.1}(\boldsymbol{I})$, $\sigma_p = 5$ and $\mathbf{\Gamma}_{0.25}(\boldsymbol{I})$). Resolution of the maps provided by T-ReX is 250 Kpc h^{-1}. Shaded blue and orange areas highlight some differences between results discussed in Sect. 6.1.1.

paths identified by DisPerSE are not relevant for minimising the total distance and, thus, they do not appear as possible paths in any of the realisations. When comparing two conservative cases, namely, $\mathbf{\Gamma}_{0.25}(\boldsymbol{I})$ and the 5σ DisPerSE persistence skeleton (lowest right panel of Fig. 13), we see that the T-ReX pattern preserves more small-scale structures and provides some paths that seems coherent with the subhalo distribution but which are not identified with the chosen parameters for the DisPerSE output (see blue shaded regions in Fig. 13).

6.1.2. Sparse data point distribution

In order to explore the robustness of the method against the datapoint density used for ridge detection, we reduce the number of subhalos in the initial dataset by keeping only those with a mass $M \geq M^{\rm cut}$. In practice, we investigate how the original filamentary map is spatially close to the recovered ones when $M^{\rm cut}$ varies. Figure 14 shows probability maps obtained for increasing values of $M^{\rm cut}$ leading to sparser and sparser input (100%,

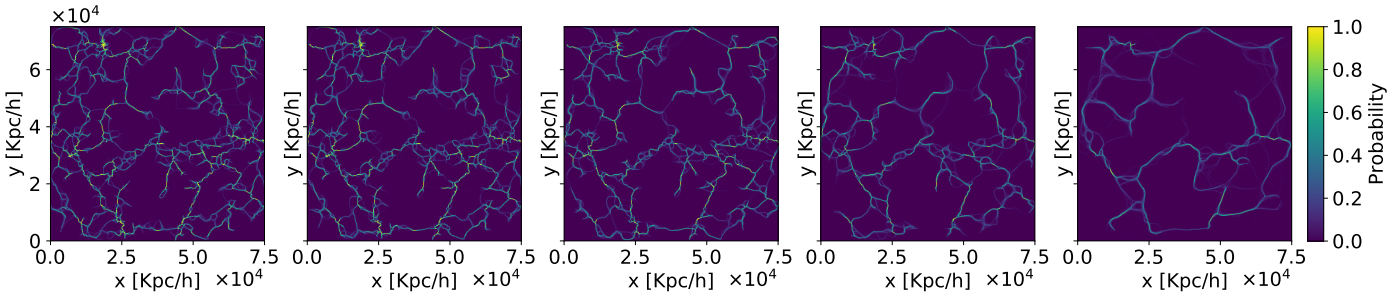

Fig. 14. Probability maps with increasing mass threshold $M^{\rm cut}$. *From left to right*, $M^{\rm cut} = \{0, 0.85, 1.35, 3.22, 11\} \times 10^{10}\, M_\odot\, {\rm h}^{-1}$ corresponding, respectively, to 100%, 83%, 60%, 31%, and 10% of the total subhalos in the slice.

83%, 60%, 31% and 10% of the initial subhalos in the slice respectively corresponding to $M^{\rm cut} = \{0, 0.85, 1.35, 3.22, 11\} \times 10^{10}\, M_\odot/h$). Visually, probability maps show a nice stability, even when the sparsity is high: patterns are pretty much the same when we keep at least 60% of the most massive objects hence recovering the essential part of the structure.

Figure 15 emphasises the spatial proximity between the different maps by representing, for each $\boldsymbol{I}_J$, where J denotes the fraction of galaxies we kept to compute the map, the cumulative distribution of $\{d_x^J\}_{x\in\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_{100})}$ defined, for a position x in the set $\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_{100})$, as

$$d_x^J = \min_{x'\in\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_J)} \|x - x'\|_2. \tag{15}$$

Hence d_x^J corresponds to the closest distance from a position x in the original skeleton obtained by keeping all subhalos, namely $\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_{100})$, to a given thresholded map $\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_J)$. This way, the distribution of d_x^J measures how far the original pattern is from the one obtained with J% of the data points.

In more than 95% of the cases, the original pattern finds a closest point in the 83% and the 60% maps at less than $1.8\,{\rm Mpc\,h^{-1}}$, showing that structures found in the three maps are spatially close and about the thickness of typical filaments (Cautun et al. 2014). When $M^{\rm cut}$ increases, the filamentary pattern traces the most prominent parts of the structure with a loss of some small scales and hence highlights coarser and coarser structures. Even though the pattern is rough with only 31% of the data points used, we still observe a nice correlation with previous maps highlighting coherent structures with 90% of the original pattern being retrieved at less than $3\,{\rm Mpc\,h^{-1}}$. As expected, an unrealistic scenario where we use only 10% of the data points associated with the most massive subhalos degrades the reconstruction of the filamentary pattern. Yet, the recovered structures show a coarse but coherent connectivity between regions. This illustrates the ability of T-ReX to recover the underlying structure with high stability with respect to deformation of the input distribution of data points.

6.2. Application to 3D data

In this section, we apply T-ReX on the 3D distribution of halos obtained from a Gadget-2 simulation (see Sect. 2) and compare our results with some other existing procedures that have also been run on the same dataset. Although the original review (Libeskind et al. 2017) considers over a dozen different methods, we focus the comparison on three procedures, namely Nexus+, DisPerSE and Bisous, so that we have a broad set of different methods using respectively scale-space representation, topological considerations, or stochastic approach to recover the filamentary pattern. Nexus+ (Cautun et al. 2013) is a classification

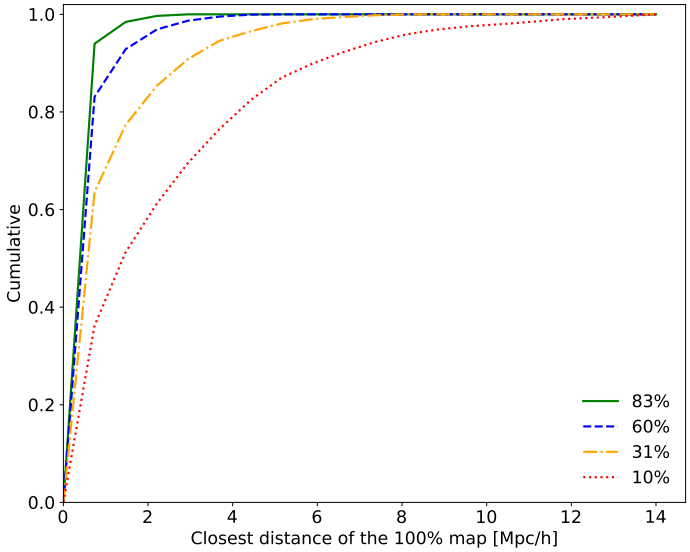

Fig. 15. Cumulative distribution of distances $\{d_x^J\}$ (see Sect. 6.1.2) between positions of the binary maps $\boldsymbol{\Gamma}_{0.25}(\boldsymbol{I}_{100})$ obtained with increasing mass threshold $M^{\rm cut}$ to the one with J% of the data points. $M^{\rm cut} = \{0.85, 1.35, 3.22, 11\} \times 10^{10}\, M_\odot\, {\rm h}^{-1}$ leading respectively to 83%, 60%, 31% and 10% of the total number of subhalos in the slice.

algorithm inspired by image processing and based on filtering techniques leading to state-of-the-art environment classification able to identify clusters, filaments and walls. The main idea is to assume that the local morphology of the density field fully encodes the environmental information. Eigenvalues of the Hessian of the density field are thus used to compute an environmental signature in each voxel of the smoothed field. The key ingredient is to compute this signature for a set of smoothed fields with a log-Gaussian filter over a range of different scales to highlight structures of different sizes. Physically motivated criteria are then used to threshold signature values and attribute a classification to each volume element. Bisous (Stoica et al. 2007) is a publicly available[5] stochastic method based on halos positions aiming at identifying the filamentary structure using a set of random parametric cylinders. Filaments are modelled as aligned and contiguous small cylinders of a given size in the galaxy distribution. The Bisous model generates two maps allowing to extract filaments spine; one characterising the probability to find a filament at a given position called the visit map and an other one corresponding to the filament orientation field. This way, spines are defined as dense regions and are aligned with the axis of the different cylinders.

[5] https://www.ascl.net/1512.008

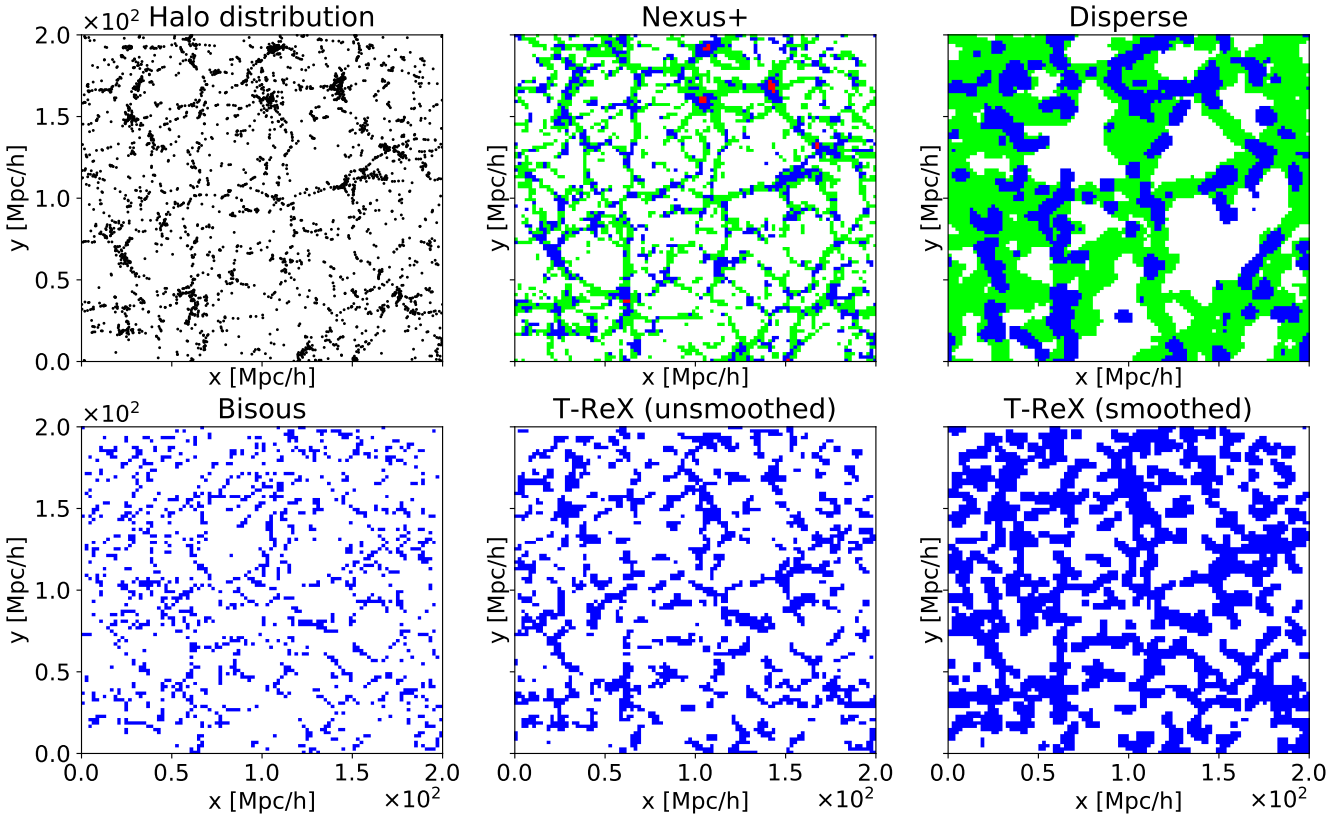

Fig. 16. Identification results provided by four detection methods on a randomly chosen $2\,\mathrm{Mpc\,h^{-1}}$ depth slice of the full 3D detection for each method. Green pixels are walls, blue are filaments, red are clusters and white are voids or unclassified regions.

We note that not only do these methods have very different mathematical definitions for what they all call clusters, filaments, and walls, but they also have been run with different input, using either DM particles or halos.

We applied T-ReX to the full halo distribution of the 3D simulated box (281465 halos in total) and built a $100 \times 100 \times 100$ grid map like other methods. For T-ReX, this means that the final probability map is computed over a 100^3 grid in which all visited voxels are considered part of the filamentary structure. As T-ReX is using 1D objects (segments of the RMST) sampled over the input space, it is preferable, for illustration and comparison, to give its filamentary pattern a "thickness" by smoothing the obtained probability map. Whenever a voxel is classified as part of the filamentary structure, a smoothing is thus performed over its 26 direct neighbors. In what follows, we call this version T-ReX$_s$ while the original result is referred to as T-ReX$_{us}$.

For illustration, following Libeskind et al. (2017), we show in Fig. 16 the results of the classification provided by each method for a $2\,\mathrm{Mpc\,h^{-1}}$ depth slice from which FoF halos were extracted (top left panel of Fig. 16). We note that all methods have been run over the full 3D cube and this is a projected slice of the detection. It is also worth noting that T-ReX identifies the filamentary pattern as a whole and does not classify the environment into clusters, filaments and walls as Nexus and DisPerSE do. To perform the comparison, we must look at the full pattern provided by each method and compare it with our extracted skeleton. We observe that T-ReX provides a satisfactory connectivity of the halos through the slice. In its smoothed version, it leads to thicker filaments compared to the results of Nexus+ and Bisous but thinner ones than Disperse, and retrieves most of the structures (filaments, walls, and clusters) obtained by the Nexus+ algorithm.

Even though these methods have been developed with different approaches, it is interesting to see whether they agree or not in the detection of the filamentary pattern. To do so in a quantitative way, we could use the proximity measurement of Eq. (15) but as the resulting patterns are presented on a $2\,\mathrm{Mpc\,h^{-1}}$ grid, the distance between them would not be accurate. Hence, we introduce a similarity measurement as follows: considering the answers provided by two detection methods, H_1 and H_2, such that $H_\bullet(x) = 1$ if the position x is part of the filamentary structure and 0 otherwise, the similarity measurement is defined as:

$$\mathcal{S}(H_1, H_2) = \frac{|H_1 \cap H_2|}{|H_1|}, \tag{16}$$

where $|H_i|$ denotes the cardinal of H_i defined as $\sum_x 1_{H_i(x)=1}$ and $|H_1 \cap H_2|$ is the cardinal of the intersection between H_1 and H_2 detections defined as $\sum_x 1_{H_1(x)=1} 1_{H_2(x)=1}$. Hence, $\mathcal{S}(H_1, H_2)$ measures the proportion of H_1 detections that are contained in H_2 and is thus asymmetric. In other words, if we consider H_2 as a reference, $\mathcal{S}(H_1, H_2)$ represents the proportion of true detections provided by H_1. Of course, such a simple metric does not provide the full information on the similarity between the considered patterns. This measure must then be completed in tandem with others, or with visual inspection, as we have done here.

Table 2 shows the similarity indices between all considered methods for the entire 3D cube. We observe that 85% of the detections provided by the unsmoothed version of T-ReX are contained in the Nexus+ skeleton and 81% of the Nexus+ detections are found by the smoothed version of T-ReX. This indicates that the smoothed version of T-Rex contains a large part of the Nexus+ skeleton but with a larger amount of the volume detected, explained by the smoothing leading to a thicker filamentary pattern. The same tendency is observed concerning

Table 2. Index of similarity $\mathcal{S}(H_1, H_2)$ as defined in Eq. (16) between the considered methods applied on the entire 3D cube.

	H_2				
H_1	T-ReX$_{us}$	T-ReX$_s$	Nexus+	DisPerSE	Bisous
T-ReX$_{us}$	1	1	0.85	0.62	0.37
T-ReX$_s$	0.48	1	0.62	0.62	0.24
Nexus+	0.53	0.81	1	0.62	0.30
DisPerSE	0.22	0.46	0.35	1	0.12
Bisous	0.66	0.87	0.86	0.62	1

Notes. T-ReX$_{us}$ refers to the unsmoothed version of the detection while T-ReX$_s$ refers to the smoothed one over the 26 neighboring voxels.

Bisous for which the detections are mostly contained in other skeletons (last row of Table 2) but not reciprocally (last column of Table 2). This is due to the sparse and unconnected detection provided by the Bisous method. The thick skeleton of DisPerSE also tends to contain a large fraction of other skeletons (fourth column of Table 2) but it fills so much volume, which is not contained in the latter (fourth line of Table 2).

7. Conclusion

In this paper, we present T-ReX, a graph-based algorithm aimed at an automatic retrieval of the underlying density from a discrete set of points. We show that it can be used to uncover the natural filamentary pattern of the Cosmic Web from a 2D or 3D galaxy distribution. The key idea of T-ReX is to find a set of centroids paving a given set of data points in its ridges by enforcing a predefined topology. To do so, the minimum spanning tree is computed over those centroids, which are iteratively moved to obtain a smoothed version of the MST. To characterise the reliability of the underlying filamentary structure, a without replacement bootstrap is used where several regularised MST are computed over a subset of data points chosen randomly and uniformly. In this way, we can build a probability map of those realisations to get the most frequent paths and highlight some regions as being part of the underlying filamentary pattern with high reliability.

For the sake of simplicity and because this topology is, at first, a fitting representation of the filamentary structure, we chose the tree topology for the centroids to highlight ridges of the point cloud distribution of galaxies. In addition, the MST provides a natural way to connect observed data points with the possibility to infer the underlying filamentary pattern by minimising the total distance linking them. However, the presented framework (see Sect. 3.3) is more general and can use any kind of graph construction. Hence, it could be interesting to investigate other topologies and in other contexts that detecting ridges. In particular, its nearest neighbors have been recently applied in several cosmological studies, such as Coutinho et al. (2016), to find new metrics characterising the Cosmic Web using graphs. Also, studying the properties of the regularised tree representation in the same way as it is done for the usual MST (see e.g. Colberg 2007; Naidoo et al. 2020) could be of interest.

In this paper, we mainly focus the application of the procedure on simulated datasets. When dealing with real data, in addition to the mathematical considerations of defining and extracting the filamentary pattern, we face the usual technical issues of observed data: noise, outliers, uneven distribution of the samples, sparsity of the representation, selection, and observational effects. Even though we showed some robustness of the estimate to noise and outliers, the ability of minimum spanning tree methods to get rid of observational (redshift-space distortions) and selection effects (missing parts of the sky) in real cosmological surveys could be a consideration of further studies.

Acknowledgements. The authors thank Pr. Einasto for helpful comments that improved the quality of the paper. We are grateful to the scikit-learn (Pedregosa et al. 2011) team for making easier and sometimes straightforward the implementation and testing of parts of the algorithm. TB thanks Pr. S. White for fruitful discussions. The authors also thank all members of the ByoPiC team (https://byopic.eu/team) for useful comments and discussions. This research was supported by funding for the ByoPiC project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant number ERC-2015-AdG 695561.

References

Alpaslan, M., Robotham, A. S., Driver, S., et al. 2014a, MNRAS, 438, 177
Alpaslan, M., Robotham, A. S., Obreschkow, D., et al. 2014b, MNRAS, 440, 1
Aragon-Calvo, M. A., Jones, B. J. T., van de Weygaert, R. M. J., & der Hulst, V. 2007, A&A, 474, 315
Aragon-Calvo, M., Weygaert, R. V. D., & Jones, B. J. T. 2010a, MNRAS, 408, 2163
Aragón-Calvo, M. A., Platen, E., Van De Weygaert, R., & Szalay, A. S. 2010b, ApJ, 723, 364
Barrow, J. D., Bhavsar, S. P., & Sonoda, D. H. 1985, MNRAS, 216, 17
Bezdek, J. C. 1981, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum pre edition (Kluwer Academic Publishers), 267
Bishop, C. M., & Svensén, M. 1998, Neural Comput., 10, 215
Bond, J. R., Kofman, L., & Pogosyan, D. 1996, Nature, 380, 603
Bonjean, V., Aghanim, N., Salomé, P., Douspis, M., & Beelen, A. 2018, A&A, 609, A49
Boruvka, O. 1926, Práce Moravské přírodovědecké společnosti, 3, 37
Bos, E. G., Van De Weygaert, R., Kitaura, F., & Cautun, M. 2014, Proc. Int. Astron. Union, 11, 271
Cautun, M., van de Weygaert, R., & Jones, B. J. 2013, MNRAS, 429, 1286
Cautun, M., Weygaert, R. V. D., Jones, B. J. T., & Frenk, C. S. 2014, MNRAS, 441, 2923
Chen, Y. C., Ho, S., Freeman, P. E., Genovese, C. R., & Wasserman, L. 2015, MNRAS, 454, 1140
Codis, S., Pogosyan, D., & Pichon, C. 2018, MNRAS, 479, 973
Colberg, J. M. 2007, MNRAS, 375, 337
Coutinho, B. C., Hong, S., Albrecht, K., et al. 2016, ArXiv e-prints [arXiv:1604.03236]
de Graaf, A., Cai, Y.-C., Heymans, C., & Peacock, J. A. 2019, A&A, 624, A48
Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977, J. R. Stat. Soc., 39, 1
Dietrich, J. P., Werner, N., Clowe, D., et al. 2012, Nature, 487, 202
Doroshkevich, A. G., & Shandarin, S. F. 1978, Sov. Astron., 22, 653
Dubois, Y., Pichon, C., Welker, C., et al. 2014, MNRAS, 444, 1453
Durbin, R., & Willshaw, D. 1987, Nature, 326, 14
Eckert, D., Jauzac, M., Shan, H., et al. 2015, Nature, 528, 105
Edelsbrunner, H., Letscher, D., & Zomorodian, A. 2002, Discrete Comput. Geom., 28, 511
Einasto, J., Joeveer, M., & Saar, E. 1980, MNRAS, 193, 353
Epps, D., & Hudson, M. J. 2017, MNRAS, 468, 2605
Forman, R. 1998, Adv. Math., 145, 90
Genovese, C. R., Perone-Pacifico, M., Verdinelli, I., & Wasserman, L. 2014, Ann. Stat., 42, 1511
Gheller, C., & Vazza, F. 2019, MNRAS, 486, 981
Gheller, C., Vazza, F., Br, M., et al. 2016, MNRAS, 462, 448
Gorban, A., & Zinovyev, A. 2005, Computing, 75, 359
Gouin, C., Gavazzi, R., Codis, S., et al. 2017, A&A, 605, A27
Hastie, T., Stuetzle, W., Hastie, T., & Stuetzle, W. 1989, J. Am. Stat. Assoc., 84, 502
Hébert-Dufresne, L., Grochow, J. A., & Allard, A. 2016, Sci. Rep., 6, 1
Jasche, J., & Wandelt, B. D. 2013, MNRAS, 432, 894
Joeveer, M., Einasto, J., & Tago, E. 1978, MNRAS, 185, 357
Kitaura, F.-S. 2013, MNRAS, 429, L84
Kraljic, K., Davé, R., & Pichon, C. 2020, MNRAS, 237
Kullback, S., & Leibler, R. 1951, Ann. Math. Stat., 22, 79
Kuutma, T., Tamm, A., & Tempel, E. 2017, A&A, 600, L6
Laigle, C., Pichon, C., Arnouts, S., et al. 2018, MNRAS, 474, 5437
Leclercq, F., Lavaux, G., Jasche, J., & Wandelt, B. 2016, J. Cosmol. Astropart. Phys., 2016, 1
Libeskind, N. I., van de Weygaert, R., Cautun, M., et al. 2017, MNRAS, 473, 1195
Lurie, J. 1999, ACM SIGACT News, 30, 14

Macqueen, J. 1967, Math. Rev., 281
Malavasi, N., Arnouts, S., Vibert, D., et al. 2017, MNRAS, 465, 3817
Malavasi, N., Aghanim, N., Tanimura, H., Bonjean, V., & Douspis, M. 2020, A&A, 634, A30
Mao, Q., Li, W., Ivor, W. T., & Sun, Y. 2016, ArXiv e-prints [arXiv:1512.02752v2]
Mao, Q., Yang, L., Wang, L., Goodison, S., & Sun, Y. 2015, Proc. SIAM Int. Conf. Data Min., 792
Martinez, H., Muriel, H., & Coenda, V. 2016, MNRAS, 445, 127
Moccia, S., Momi, E. D., Hadji, S. E., & Mattos, L. S. 2018, Comput. Methods Programs Biomed., 158, 71
More, S., Kravtsov, A. V., Dalal, N., & Gottlöber, S. 2011, ApJS, 195, 4
Naidoo, K., Whiteway, L., Massara, E., et al. 2020, MNRAS, 491, 1709
Nicastro, F., Kaastra, J., Krongold, Y., et al. 2018, Nature, 558, 406
Pedregosa, F., Weiss, R., & Brucher, M. 2011, J. Mach. Learn. Res., 12, 2825
Qiu, X., Qi, M., Ying, T., et al. 2017, Nat. Methods, 14, 979
Roweis, S. T., & Saul, L. K. 2000, Science, 290, 2323
Sarron, F., Adami, C., Durret, F., & Laigle, C. 2019, A&A, A49
Schaap, W., & Weygaert, R. 2000, A&A, 363, L29
Silverman, B. 1986, Monographs on Statistics and Applied Probability
Smola, A. J., Mika, S., Sch, B., & Williamson, R. C. 2001, J. Mach. Learn. Res., 1, 179
Sousbie, T. 2011, MNRAS, 414, 350
Springel, V., Wang, J., Vogelsberger, M., et al. 2008, MNRAS, 391, 1685
Springel, V., White, S. D. M., Jenkins, A., et al. 2005, Nature, 435, 629
Stoica, R. S., Martínez, V. J., & Saar, E. 2007, J. R. Stat. Soc. Ser. C: Appl. Stat., 56, 459
Tanimura, H., Hinshaw, G., McCarthy, I. G., et al. 2019, MNRAS, 483, 223
Tanimura, H., Aghanim, N., Bonjean, V., Malavasi, N., & Douspis, M. 2020, in press, https://doi.org/10.1051/0004-6361/201937158
Tibshirani, R. 1992, Stat. Comput., 2, 183
Tibshirani, R., Walther, G., & Hastie, T. 2001, J. R. Stat. Soc. Ser. B: Stat. Method., 63, 411
Vogelsberger, M., Genel, S., Springel, V., et al. 2014, MNRAS, 444, 1518
York, D. G., Adelman, J., Anderson, J., et al. 2000, ApJ, 120, 1579
Yuille, A. L. 1990, Neural Comput., 2, 1
Zel'dovich, Y. a. B. 1970, A&A, 500, 13